PROJECT REPORT ON

"The Skills Gap Bridge: A Course Recommendation System for Optimal Student Placement"

A report submitted in partial fulfillment of the requirements for

INTEGRATED PROFESSIONAL CORE COURSE- FUNDAMENTALS OF MACHINE LEARNING (AD2001-1)

Submitted by

Narasimha M Pai	**NNM22AD034**
Sanidhya K Bhandary	**NNM22AD046**
Shetty Khyathi Rajesh	**NNM22AD050**
Sujan U Kulai	**NNM22AD057**

Under the Guidance of

Dr. Venugopala P S

Professor & Head of Department

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

CERTIFICATE

This is to certify that the Integrated Professional Core Course project entitled "The Skills Gap Bridge: A Course Recommendation System for Optimal Student Placement" report has been submitted by Narasimha M Pai(NNM22AD034), Sanidhya K Bhandary(NNM22AD046), Shetty Khyathi Rajesh(NNM22AD050) and Sujan U Kulai(NNM22AD057) of II year B. Tech, bonafide students of NMAM Institute of Technology, Nitte, has completed the project work during the academic year 2023-2024 fulfilling the partial requirements of Integrated Professional Core Course lab in Artificial Intelligence & Data Science at NMAM Institute of Technology, Nitte.

Project Guide	**Course Coordinator**	**HOD**

Table of contents

Chapter 1

Abstract

The ever-evolving job market demands a skilled and adaptable workforce. However, the traditional academic curriculum may not always align perfectly with industry needs. This disconnect creates a "skills gap" hindering student employability. This project proposes a solution by leveraging machine learning to bridge this gap and streamline the student-to-workforce transition.

We present a novel course recommendation system that analyzes both a student's existing skillset and the critical skills sought after by potential employers. This analysis utilizes machine learning techniques to identify skill gaps between a student's profile and their desired career paths. By pinpointing these gaps, our system recommends relevant courses that strategically address them.

This project offers a win-win scenario for both students and employers. Students gain valuable guidance in selecting courses that directly enhance their employability for their chosen field. Employers, on the other hand, benefit from a more qualified talent pool with the specific skillsets required for their positions.

This report details the methodology employed throughout the project's development. We delve into the techniques used for student skill identification and the extraction of critical skills from job postings. The core of the system, the machine learning model, will be thoroughly explained, alongside its role in recommending courses to bridge identified skill gaps.

Furthermore, the report explores the potential impact this system could have on educational institutions by providing valuable insights into aligning curriculum with market demands. We conclude by discussing exciting future directions for development that can further enhance the system's effectiveness and expand its reach.

Chapter 2

Introduction

2.1. Description of the problem being addressed:

This project utilizes machine learning to bridge the gap between student skills and industry demands. By analyzing student skillsets and job posting requirements, an ML model recommends courses that address skill gaps and enhance student employability. This system benefits both students, who can make informed course choices, and employers, who gain access to a more qualified talent pool.

2.2. Objectives of the project:

2.2.1. Enhance Student Employability:

By recommending courses that align with industry needs, the project aims to equip students with the skills most sought after by employers, making them more competitive in the job market.

2.2.2. Bridge Skills Gap:

Analyze the skill sets of students and the skill requirements of job postings to identify areas where students lack necessary skills. The recommended courses will target these gaps, ensuring a smoother transition from academia to the workforce.

2.2.3. Improve Student Course Selection:

The project empowers students to make informed decisions about their coursework by providing data-driven recommendations that enhance their career prospects.

2.3. Importance of the project in the context of machine learning:

This project holds significant importance within the realm of machine learning due to its focus on bridging the often-present gap between academic programs and industry demands. The ever-evolving job market necessitates a dynamic approach to education, and machine learning offers a powerful tool in this regard. By analyzing vast amounts of data on student skills and employer needs, the project utilizes machine learning to create personalized course recommendations. This not only empowers students to make informed choices about their education but also fosters a more qualified workforce, ultimately benefiting both students and employers. Furthermore, the project contributes valuable data and insights into the field of educational machine learning, paving the way for further development of intelligent systems that can personalize and optimize learning experiences.

Chapter 3

Literature Survey

Learning Path Recommendation:
With the application of recommendation technology in the field of e-learning, the development of the research field of personalized learning recommendation has been promoted. Personalized learning recommendation is a key link in achieving personalized learning service. Its objective is to provide individual learners with learning resources that meet their learning needs to enhance learning experience and effects. In the cur-rent related research, the objects of personalized learning recommendation involve learning resources such as exercises, courses, friends, and learning paths. Among them, learning path recommendation is one of the hotspots of personalized learning recommendations research. The purpose of learning path recommendation is to help individual learners achieve their learning goals; just as the following suggest to the learner's characteristics and needs, personalized learning path is recommended. Some learning content sequence (learning object, concept, and course) that matches their prerequisites can then be called a learning path. At present, some researchers have produced many different personalized learning path recommendation methods through various knowledge models, algorithms, and technologies.

In the research of learning path recommendation, parameters are generally used to determine the personalized characteristics and needs of users. These parameters also describe different learning scenarios, besides describing the learning needs of users and their different characteristics, such as learning style and knowledge background. Through literature research, it has been found that the researchers consider different personalized parameters to recommend personalized learning path. Moreover, some existing data mining methods use clustering to combine the characteristics of historical learners. Some studies use intelligence and optimization algorithms to combine the learning effect of historical learners. These include genetic algorithm, ant colony algorithm (ACO), and neural network which have been used by researchers to solve learning path generation and recommendation. In, ACO-based learning path planning technology treated a learner as ant-like agent. It first calculated the pheromone in the entire content network and then recommended a learning path to a new student. Research by used the concept familiarity of all former learners in the same cluster in the learning path discovery algorithm, together with ant colony algorithm (ACO) to generate learning paths and recommend learning paths to new learners. In recent years, neural networks have been the most used method for learning path prediction in the field of learning path recommendation. In, a personalized learning full-path recommendation model was proposed based on LSTM neural network, which relied on clustering and machine learning technology. First, a group of learners was clustered based on their feature similarity measurement, and a long short-term memory (LSTM) model was trained to predict the learning path and learning effect. The personalized learning full path was then selected from the path prediction result. A method of constructing a learning path recommendation system was proposed using ability graphs. The ability graphs were used to display user's scores. Recursive neural networks were used to construct sequence-based prediction models to predict and recommend next questions that should be learned by the user.

Chapter 4

Problem Statement

The education system and the job market often operate in silos, leading to a critical misalignment in the skills students develop and the skills employers require. This "skills gap" creates a double-edged sword. On one hand, graduates struggle to find employment because their skillsets don't match industry needs. On the other hand, companies face difficulties filling open positions due to a lack of qualified candidates with the relevant skills. This situation necessitates a bridge to connect these two worlds, ensuring students graduate with the necessary skills for a smooth transition into the workforce and fulfilling the talent needs of businesses.

Chapter 5

Dataset Description

5.1. Description of the dataset used in the project:

The dataset comprises information about students and companies like student name,student's technical skills, soft skills, cgpa and company name and skills required This dataset is used in the project to explore the relationship between student attributes (technical skills, soft skills, CGPA) and their suitability for different companies. By analyzing this data, the project aims to develop insights and potentially predictive model to determine whether a student is a good fit for a given company and a recommendation system to recommend courses to be a good fit for their chosen company.

5.2.Source of the dataset:

5.2.1. Student Details: Information regarding student names, CGPA and skillsets was obtained directly from students themselves. This was achieved through asking the students directly for their information.
5.2.2. Company Details: Data on company names were obtained from the official college website and the skillsets were obtained from internet from the official website of each company.

5.3. Features included in the dataset:

Student:
* Student Name: This attribute identifies the student.
* CGPA: This numerical value represents the student's Cumulative Grade Point Average, offering a general indicator of academic performance.
* Student Skills: This attribute captures the skills possessed by the student. Each skill is listed as a separate keyword (Python, Java, Web Dev, SQL, Excel, Cybersecurity, PowerBI).
Company:
* Company Name: This attribute identifies the company that posted the job opening.
* Skillset for company: This attribute specifies the skills companies are seeking in potential employees. Similar to student skills, each skill is listed as a separate keyword (Python, Java, Web Dev, SQL, Excel, Cybersecurity, PowerBI).

5.4. Size of the dataset

The current dataset consists of information for 72 students of AI&DS branch and same number of company details.

5.5. Preprocessing steps applied to the dataset

1. Skill Assignment and Comparison:
Random Skill Assignment
Skill Numerical Encoding
Skillset Comparison.
2. Data Transformation:
Binary Outcome Creation
CSV File Generation

Chapter 6

Methodology

6.1. Description of the preprocessing:

6.1.1. Skill Assignment and Comparison:

Random Skill Assignment: The random() function was used to randomly assign skills (e.g., Python, Java) and soft skills(e.g., Teamwork, Communciation) to each student in the dataset.

Skill Numerical Encoding: Each skill was assigned a unique numerical value (e.g., Python: 1, Java: 2).

Skillset Comparison: Student and company skill sets were compared by iterating through each skill and checking if the student possessed the required skill (as indicated by the corresponding numerical value present in the company's skill set).

2. Data Transformation:

Binary Outcome Creation: The comparison results ("Yes" for being fit for the company, "No" for not) were converted into binary format (1 for being fit for the company, 0 for not).

CSV File Generation: The processed data, including student skills, company skills, and the binary comparison outcome for each skill pair, was saved as a CSV file using the pandas library.

6.2. Description of the machine learning techniques/algorithms/models used:

K-Means Clustering: We applied the K-Means clustering algorithm to group students into three clusters based on their CGPA (Cumulative Grade Point Average). This technique identifies groups (clusters) with similar CGPA values, allowing for targeted analysis.

Scatter Plot Visualization: A scatter plot was created to visualize the relationship between student skills and their respective CGPA clusters. This visualization provided insights into potential skill variations across different academic performance levels.

Rule-Based Matching: We implemented a rule-based system to compare student skills with company skill requirements. Individual if-else statements determined whether a student possessed each required skill for a specific company. This approach offered a clear and interpretable way to assess student-company skill alignment.

Eligible Company List Generation: Based on the skill matching results, a new dataset was created. This dataset contained student names and all the companies they were eligible for, considering their skills.

Decision Tree Model: A decision tree algorithm was employed to analyze the generated dataset containing student-company eligibility information. Decision trees excel at handling discrete data (e.g., skills possessed/not possessed) and provide a clear interpretable model for understanding the reasoning behind recommendations.

Confusion Matrix Evaluation: To assess the performance of the decision tree model, a confusion matrix was generated. This matrix visualizes the number of correct and incorrect predictions made by the model, providing valuable insights into its accuracy and potential for improvement.

Recommendation System using knn: This recommendation system utilizes K-Nearest Neighbors (KNN) to identify companies with similar skill requirements to the user's target company. KNN analyzes the skillsets demanded by various companies and recommends positions for which the user's skills closely match companies with overlapping needs.

6.3. Justification for the chosen techniques/algorithms/models:

K-Means clustering is a simple and efficient unsupervised learning algorithm, making it ideal for exploratory data analysis when initial group structures are unknown. In this case, we aimed to identify potential subgroups within the student population based on their CGPA.

Visualization with scatter plots provides a clear visual representation of skill distribution across CGPA clusters. This helps in understanding potential correlations or trends between academic performance and skill sets.

A rule-based approach with conditional statements offers a straightforward way to compare student skills and company requirements. It's suitable for situations where the decision-making process is well-defined and relatively simple.

Decision tree was used for creating a new dataset based on the matching results which helps in streamlining the data for subsequent analysis and model training.

KNN is well-suited for our recommendation system because it excels at finding similar items (companies) based on user input (skills), making it effective for suggesting relevant opportunities and identifying skill gaps.

Chapter 7

Implementation

7.1. Details of how the methodology was implemented:

a. K-Means Clustering:
Purpose: Group students into clusters based on their CGPA (Cumulative Grade Point Average).
Implementation:
Extracted CGPA values from the DataFrame.
Reshaped the data into a 2D array for KMeans compatibility.
Defined the number of clusters (k) as 3.
Created a KMeans instance with the chosen number of clusters.
Fitted the model to the CGPA data.
Obtained the cluster labels assigned to each student.
Added a new column named "Cluster" to the DataFrame containing these labels.
Visualized the clustering results using a scatter plot with CGPA on the x-axis and cluster labels on the y-axis.

b. Scatter Plot Visualization:
Purpose: Visually explore the relationship between student skills and their CGPA clusters.
Implementation:
Created a scatter plot with df['Cluster'] on the x-axis and df['Skill_label'] (assuming this represents encoded skills) on the y-axis.
Added labels and a title to the plot for better interpretation.

c. Rule-Based Matching (not explicitly shown in code):
Purpose: Determine whether a student possesses each required skill for a specific company.
Implementation:
Use conditional statements (if-else) to compare student skills with company requirements for each skill.
Create a new dataset indicating whether a student is eligible for a company based on skill matching.

d. Decision Tree Model:
Purpose: Analyze student-company eligibility information (generated from rule-based matching) and predict company fit (Yes/No) for students.
Implementation:
Split the generated student-company eligibility dataset into features (e.g., student ID, company name, skills) and target variable (company fit - Yes/No).
Trained a DecisionTreeClassifier model on the training data.
Evaluated model performance using metrics like accuracy, precision, recall, and F1-score.
Analyzed the decision tree structure and predictions to understand the factors influencing company fit recommendations.

e. Confusion Matrix Evaluation:

Purpose: Assess the performance of the decision tree model.

Implementation:

Used the confusion_matrix function from sklearn.metrics to generate a confusion matrix based on the model's predictions and actual company fit values.

Visualized the confusion matrix with a heatmap using seaborn.

f. Recommendation System using KNN:

Purpose: KNN aims to find and recommend courses depending on user's and company's skillsets.

Implementation:

User skillset (comp and ski) is defined to demonstrate prediction.

The code uses the trained model (knn) to predict the most suitable course (predicted_class) for this skillset based on similar skills that led to successful course completions (or other criteria encoded in y).

It then checks the predicted class and retrieves the corresponding course name from the recom dictionary. If the prediction suggests a good fit ("Your Skill set matched..."), a success message is displayed. Otherwise, the code recommends a course based on the user's company number (comp), suggesting the skill potentially required for that course based on the training data.

7.2. Tools and libraries used for implementation:

pandas (pd): Used for data manipulation tasks like reading CSV files, creating DataFrames, adding/removing columns, and data transformations.

numpy (np): numerical computing functionalities, potentially used for operations like reshaping data for KMeans clustering.

matplotlib.pyplot (plt): Used for creating scatter plots to visualize data distributions and relationships.

seaborn (sns): (Used for confusion matrix visualization) Built on top of matplotlib, offering a higher-level interface for creating statistical graphics like heatmaps (confusion matrix).

sklearn.preprocessing.LabelEncoder: Used for encoding categorical data (e.g., skills) into numerical labels before feeding it to machine learning models.

sklearn.model_selection.train_test_split: Used to split the data into training and testing sets for model evaluation.

sklearn.tree.DecisionTreeClassifier: Used to create and train a decision tree model for predicting company fit based on student data.

sklearn.tree.export_text: Used to export the decision tree structure as text for analysis and understanding the reasoning behind its predictions.

sklearn.metrics: Provides various metrics for evaluating model performance, including accuracy_score, precision_score, recall_score, f1_score, and confusion_matrix.

sklearn.neighbors.KNeighborsClassifier: Used to build a model that classifies data points based on the similarity (distance) to their nearest neighbors in the training data.

7.3. Pseudocode:

```
# Load data
data = load_data("placementdata.csv")

# Preprocess data (encode categorical features)
preprocess_data(data)

# Separate features and target variable
features, target = separate_data(data)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = split_data(features, target)

# Train decision tree model
model = train_decision_tree(X_train, y_train)

# Make predictions on testing data
predictions = predict(model, X_test)

# Evaluate model performance (accuracy, confusion matrix)
evaluate_model(predictions, y_test)

# Visualize confusion matrix
visualize_confusion_matrix(predictions, y_test)

#Recommendation System
predicted_class = knn.predict(np.array([[comp, ski]]))[0]
if predicted_class == 0: recommend course from 'recom' using 'comp', else: print "Skills Matched"

# Data Preparation
1. Read student data from a CSV file (e.g., "placementdata.csv").
2. Extract CGPA values for K-Means clustering.

# K-Means Clustering (Analyze student performance)
3. Group students into 3 clusters (adjustable) based on their CGPA using K-Means.
4. Visualize the clusters using a scatter plot (Skills vs. cluster labels).

# Skill Analysis
5. Implement rule-based matching to compare student skills with company requirements.
6. Create a new dataset containing student-company eligibility based on skill matching.
```

Decision Tree Model (Predict company fit)

7. (Assuming data from step 6) Split data into features (student/company info) and target (company fit).
8. Train a Decision Tree model on the training data to predict company fit (Yes/No) for students.
9. Evaluate model performance using metrics like accuracy and confusion matrix.

Recommendation System using KNN

10. Analyze the decision tree structure to identify missing skills leading to "No" company fit predictions.
11. Predict the most suitable course class (predicted_class) for the user's skillset (comp, ski).
12. If the predicted class indicates a good fit (check prediction criteria), display a success message. Otherwise, recommend the course corresponding to the user's company number (comp) from the recom dictionary (suggesting a potentially missing skill).

Chapter 8

Results

8.1. Results obtained from the implementation:

a. Clustering

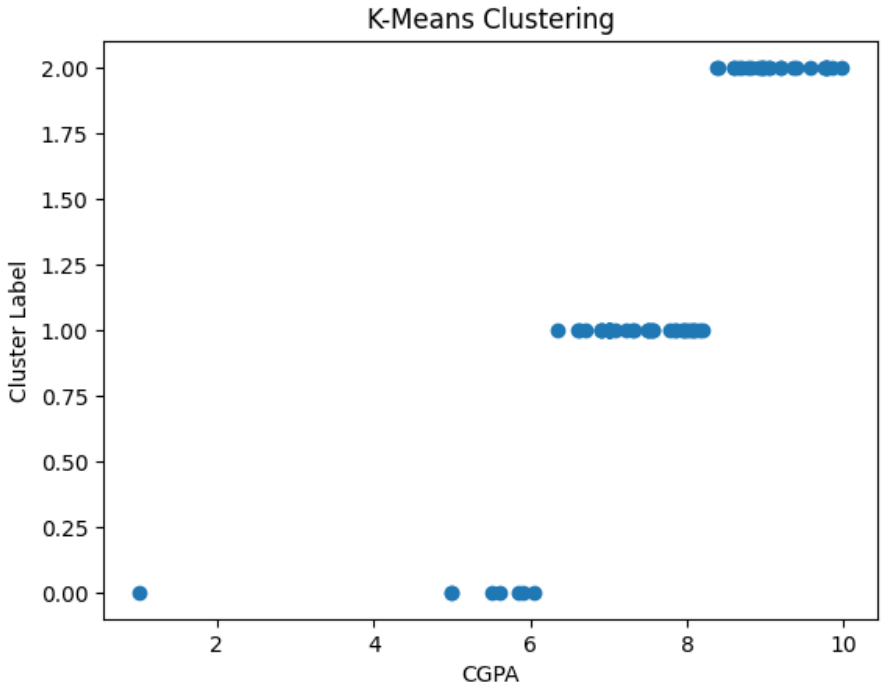

Figure 8.1: Clustering

Figure 8.1 summarizes the K-Means clustering results. Based on K-Means clustering, students were segmented into three distinct groups with varying average CGPA (e.g., high, medium, low).

b. Scatter plot

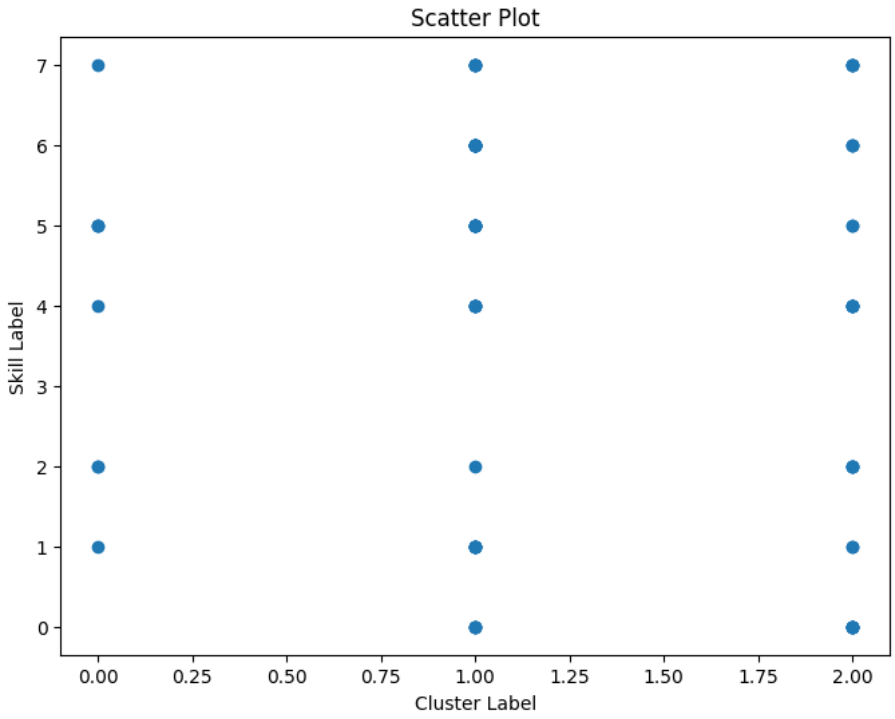

Figure 8.2: Scatter plot

Figure 8.2 shows a scatter plot with CGPA cluster on the x-axis and skills labels on the y-axis. This visually represents how students are distributed across clusters based on their skills.

c. Decision Tree

Table 8.2: Decision Tree

Metric	Value
Accuracy	0.9930555555555556
Precision	0.9811320754716981
Recall	1.0
F1-score	0.9904761904761905

Table 8.2 presents the evaluation metrics for the decision tree model. It shows the values of accuracy, precision, recall and f1-score.

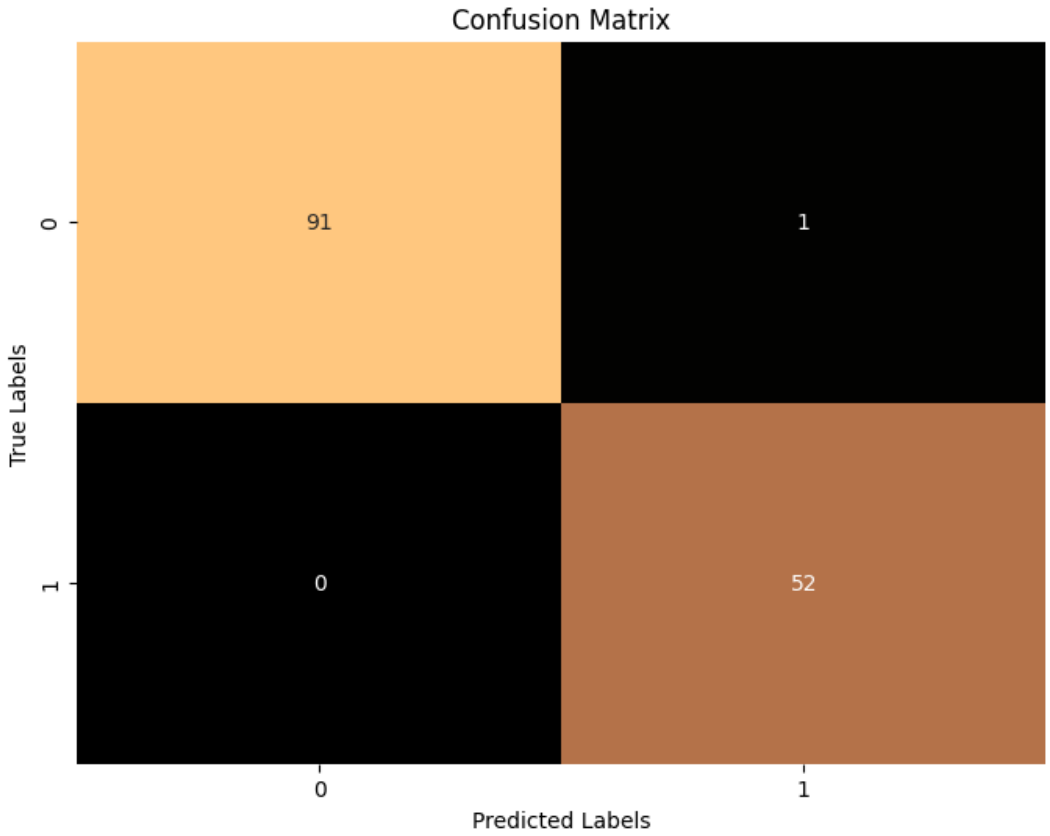

Figure 8.3: Confusion Matrix

Figure 8.3 shows the confusion matrix. It shows that 91 students with a good actual fit ("Yes") were correctly predicted as a good fit by the model ("Yes") (True Positives),1 student with a good actual fit ("Yes") were incorrectly predicted as not a good fit ("No") by the model (False Negatives), 0 students with a not-so-good actual fit ("No") were incorrectly predicted as a good fit ("Yes") by the model (False Positives), 52 students with a not-so-good actual fit ("No") were correctly predicted as not a good fit ("No") by the model. (True Negatives).

d. Recommendation System

Accuracy of k-Nearest Neighbor (k=3): 0.9166666666666666
Predicted class for the sample: 0
Your Skill didnt meet the skills required
We recommend you to Study this Course : Java

The above is an output for the student having skill as Python & choosing the company Mercedes Benz. It shows that the student's skill doesn't match company's skill. So it is recommending the course Java.

8.2. Evaluation metrics used to assess the performance of the model:

Evaluation metrics are crucial for assessing the performance of a machine learning model. They provide quantitative measures to understand how well the model performs. The common evaluation metrics are:

a. Accuracy:

It measures the overall proportion of correct predictions made by the model. In this context, it represents the percentage of students for whom the model correctly predicted their company fit ("Yes" or "No").

Formula: Accuracy = (True Positives + True Negatives) / Total Samples

For this model, we got an accuracy of 0.9930555555555556.

b. Precision:

It measures the accuracy of positive predictions (students predicted as a good fit for companies - "Yes"). It tells you how many students the model predicted as a good fit were actually a good fit for companies.

Formula: Precision = True Positives / (True Positives + False Positives)

For this model, we got a precision of 0.9811320754716981.

c. Recall:

It measures the completeness of positive predictions. It tells you how well the model identifies students who are actually a good fit for companies ("Yes"). Out of all students who were a good fit (according to actual data), how many did the model predict as a good fit?

Formula: Recall = True Positives / (True Positives + False Negatives)

For this model, we got a recall value of 1.0.

d. F1-Score:

It combines precision and recall into a single metric, providing a balanced view of model performance.

Formula: F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

For this model, we got a F1 measure as 0.9904761904761905.

Chapter 9

Conclusion

This project explored a data-driven approach to student placement, leveraging machine learning techniques to analyze student data and predict company fit. K-Means clustering provided initial insights into student performance by grouping them based on CGPA. The decision tree model, built on student and company information, aimed to predict whether a student was a suitable candidate for a particular company. The confusion matrix and evaluation metrics like accuracy, precision, recall, and F1-score were instrumental in assessing the model's performance.

By analyzing the decision tree structure, we gained valuable insights into the factors influencing company fit. This knowledge paves the way for developing a course recommendation system that addresses identified skill gaps. By recommending courses that target these missing skills, we can potentially improve student preparedness and increase their chances of landing their desired placements.

Overall, this project demonstrates the potential of machine learning for enhancing student placement strategies. By combining data analysis with decision tree models and leveraging the resulting insights, we can create a more targeted and effective system that benefits both students and companies. However, it's important to acknowledge that this is a starting point. Further refinements and exploration of other machine learning algorithms could lead to even more accurate predictions and a more robust student placement system.

Chapter 10

Reference

- Kalkar, Shailesh & Chawan, Pramila. (2022). Recommendation System using Machine Learning Techniques. 2395-0056.
- Portugal, Ivens & Alencar, Paulo & Cowan, Donald. (2015). The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review. Expert Systems with Applications. 97. 10.1016/j.eswa.2017.12.020.
- Troussas, Christos & Krouska, Akrivi. (2022). Path-Based Recommender System for Learning Activities Using Knowledge Graphs. Information. 14. 9. 10.3390/info14010009.
- Papakostas, Christos & Troussas, Christos & Krouska, Akrivi & Sgouropoulou, Cleo. (2022). Personalization of the Learning Path within an Augmented Reality Spatial Ability Training Application Based on Fuzzy Weights. Sensors. 22. 7059. 10.3390/s22187059.